(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) **Offenlegungsschrift**

(10) **DE 197 40 550 A 1**

(51) Int. Cl.[6]: **G 05 B 19/04**

(21) Aktenzeichen: 197 40 550.9
(22) Anmeldetag: 15. 9. 97
(43) Offenlegungstag: 16. 4. 98

DE 197 40 550 A 1

96P8617 DE

(66) Innere Priorität:
296 17 837.3 14. 10. 96

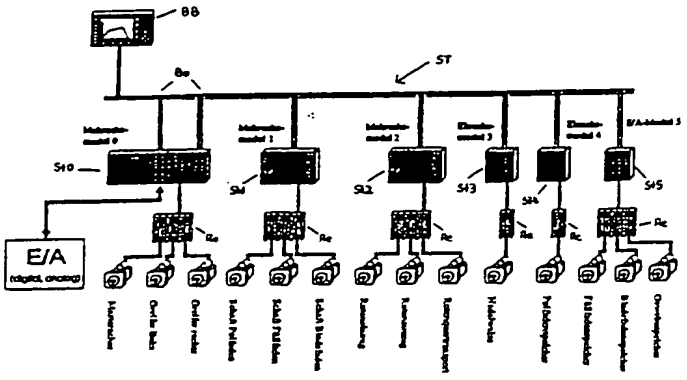
(71) Anmelder:
Siemens AG, 80333 München, DE

(72) Erfinder:
Heber, Tino, Dr.-Ing., 09599 Freiberg, DE; Kirste, Steffen, Dr.-Ing., 09120 Chemnitz, DE; Heß, Karl, Prof. Dr.-Ing.habil., 09122 Chemnitz, DE; Wucherer, Klaus, Dipl.-Ing., 90610 Winkelhaid, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Steuerung

(57) Die Erfindung betrifft eine Steuerung, welche versehen ist mit Mitteln zum Steuern eines technischen Prozesses und/oder mit Mitteln zur Steuerung der Bewegung einer Verarbeitungsmaschine und welcher ein Steuerprogramm zuführbar ist, das die Steuerung während eines Steuerbetriebs abarbeitet. Die Verwirklichung von Prozeßfunktionalitäten sowie von technologischen Bewegungsabläufen von Verarbeitungsmaschinen wird dadurch vereinfacht, daß das Steuerprogramm mit Software-Modulen versehen ist, welche mindestens eine CPU-Einheit der Steuerung während des Steuerbetriebs abarbeitet, wobei die Software-Module derart konfiguriert sind, daß diese zur Prozeßsteuerung und/oder zur Bewegungssteuerung dienen.
Die Erfindung wird angewandt bei SPS/NC-Steuerungen.



DE 197 40 550 A 1

Beschreibung

Die Erfindung betrifft eine Steuerung, welche versehen ist mit Mitteln zum Steuern eines technischen Prozesses und/oder mit Mitteln zur Steuerung der Bewegung einer Verarbeitungsmaschine und welcher ein Steuerprogramm zuführbar ist, das die Steuerung während eines Steuerbetriebs abarbeitet. Darüber hinaus betrifft die Erfindung ein Programmiergerät mit Mitteln zum Erstellen eines Steuerprogramms für eine derartige Steuerung.

Aus dem Siemens-Katalog ST 70, Ausgabe 1996, Kapitel 3, 4 und 8, ist eine speicherprogrammierbare Steuerung sowie ein Programmiergerät zum Erstellen eines Steuerprogramms für eine derartige speicherprogrammierbare Steuerung bekannt. Wesentliche Bestandteile dieser speicherprogrammierbaren Steuerung sind Baugruppen für zentrale Aufgaben (CPU-Einheiten) sowie Signal-, Funktions- und Kommunikationsbaugruppen. Die CPU-Einheit der speicherprogrammierbaren Steuerung arbeitet während des Steuerbetriebs zyklisch ein Steuerprogramm ab, welches ein Programmierer mit einem mit einem Software-Werkzeug versehenen Programmiergerät erstellt und welches zur Lösung einer Automatisierungsaufgabe vorgesehen ist. Während der zyklischen Bearbeitung liest die CPU-Einheit zunächst die Signalzustände an allen physikalischen Prozeßeingängen ab und bildet ein Prozeßabbild der Eingänge. Das Steuerprogramm wird weiter unter Einbeziehung interner Zähler, Merker und Zeiten schrittweise abgearbeitet, und schließlich hinterlegt die CPU-Einheit die errechneten Signalzustände im Prozeßabbild der Prozeßausgänge, von welchem diese Signalzustände zu den physikalischen Prozeßausgängen gelangen. Dieses Steuerprogramm umfaßt gewöhnlich Software- Funktionsbausteine, die einen Betrieb der Signal- und/oder Funktions- und/oder Kommunikationsbaugruppen ermöglichen. Eine dieser Funktionsbaugruppen in Form einer NC-Steuerungsbaugruppe ist zur Steuerung des technologischen Bewegungsablaufs einer Verarbeitungsmaschine einsetzbar. Dazu überträgt die CPU-Einheit, welche üblicherweise Prozeßsteuerungsfunktionalitäten verwirklicht, dieser NC-Steuerungsbaugruppe Parameter, z. B. Parameter in Form von Start/Stopp-Koordinaten der zu steuernden Antriebsachsen der Verarbeitungsmaschine. Ferner wählt die CPU-Einheit auf der NC-Steuerungsbaugruppe ablauffähige Verfahrensprogramme aus, die ein Prozessor der NC-Steuerungsbaugruppe zur Steuerung des Bewegungsablaufs einer Verarbeitungsmaschine abarbeitet.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, eine Steuerung der eingangs genannten Art anzugeben, welche die Verwirklichung von Prozeßfunktionalitäten sowie von technologischen Bewegungsabläufen von Verarbeitungsmaschinen vereinfacht.

Darüber hinaus ist ein Programmiergerät zu schaffen, das die Erstellung eines Steuerprogramms für eine derartige Steuerung vereinfacht.

Diese Aufgabe wird im Hinblick auf die Steuerung mit den im kennzeichnenden Teil des Anspruchs 1, im Hinblick auf das Programmiergerät mit den im kennzeichnenden Teil des Anspruchs 6 angegebenen Maßnahmen gelöst.

Vorteilhaft ist, daß Prozeßsteuerungsfunktionalitäten von an sich bekannten speicherprogrammierbaren Steuerungen (SPS) und Bewegungsfunktionalitäten von an sich bekannten NC-Steuerungen bzw. NC-Steuerungsbaugruppen in einem einheitlichen, konfigurierbaren Steuerungssystem verwirklicht werden. Dadurch können projektabhängige Steuerungen als Varianten in einer Konfigurationsphase gebildet werden und es wird vermieden, separat zur Verfügung stehende "SPS-Technik" und "NC-Technik" zu einem System zusammenzufügen.

Vorteilhafte Ausgestaltungen der Erfindung ergeben sich aus den in den Unteransprüchen angegebenen Maßnahmen.

Anhand der Zeichnung, in der ein Ausführungsbeispiel der Erfindung veranschaulicht ist, werden im folgenden die Erfindung, deren Ausgestaltungen sowie Vorteile näher erläutert.

Es zeigen:

Fig. 1 die Programmstruktur eines Software-Moduls,
Fig. 2a bis 4b Deklarationstabellen,
Fig. 5a bis 7b Bewegungsbefehlstabellen,
Fig. 8 eine Deklarationstabelle von Achsverbänden,
Fig. 9 eine Profildeklarationstabelle,
Fig. 10 eine Bewegungsattributstabelle,
Fig. 11 eine Bewegungsfunktionstabelle,
Fig. 12 eine Konfigurationselemententabelle,
Fig. 13 eine Variablendeklarationstabelle,
Fig. 14 eine Zugriffspfaddeklarationstabelle,
Fig. 15 eine Kommunikationsfunktionstabelle,
Fig. 16 den Prinzipaufbau einer Rutenwebmaschine,
Fig. 17a und 17b ein Bewegungsdiagramm einer Rutenwebmaschine und
Fig. 18 eine Steuerungsstruktur.

In Fig. 1 ist mit 1 ein Modul bezeichnet, welches im vorliegenden Beispiel zur Verwirklichung des Bewegungsablaufs einer Verarbeitungsmaschine vorgesehen ist und welches ein Programmierer auf einem hier nicht dargestellten Programmiergerät erstellt. Das Modul 1 ist Teil eines Steuerprogramms, das nach einer Übersetzung in eine geeignete Maschinensprache einer Steuerung on- oder offline in diese Steuerung übertragbar ist und das eine CPU-Einheit dieser Steuerung während des Steuerbetriebs abarbeitet. Das Modul 1 setzt sich aus einem Deklarationsteil 2, aus mindestens einem zyklischen Programm 3a, 3b und aus mindestens einem sequentiellen Programm 4a, 4b zusammen. Auf den Deklarationsteil 2 greifen alle Programme 3a, 3b, 4a, 4b des Moduls 1 zu und es sind in diesem Deklarationsteil 2 Programmnamen, Programmtypen, Variablen und/oder Datenstrukturen und/oder Bewegungsprofile hinterlegt. Die zyklischen Programme 3a, 3b sind zur Koordination der durch diese Programme 3a, 3b aufrufbaren sequentiellen Programme 4a, 4b vorgesehen. Für den Fall, daß Module zur Prozeßsteuerung vorgesehen sind, verwirklichen die zyklischen Programme derartiger Module Funktionalitäten einer speicherprogrammierbaren Steuerung. Unabhängig davon, ob die Module zur Verwirklichung von Prozeßfunktionalitäten und/oder zur Verwirklichung von Bewegungsfunktionalitäten einer Verarbeitungsmaschine dienen, arbeitet die CPU-Einheit der Steuerung diese Module ab. Innerhalb dieses Moduls 1 werden gewöhnlich lokale Variable, Eingangs- und Ausgangsvariable sowie sequentielle und zyklische Programme mit einem

Programmiergerät programmiert, konfiguriert und deklariert. Auf alle Variablen des Moduls können die zu dem Modul gehörenden Programme uneingeschränkt zugreifen. Dazu sind Deklarationsvorschriften für die Module sowie für deren Variablen vorgesehen. Beispiele von derartigen Deklarationsvorschriften sind in den Fig. 2a, 2b, 3 und 4 gezeigt, in welchen in Tabellen 1 bis 4 eine Deklaration von Modulen, von Schlüsselwörtern für die Variablen, Beispiele für eine Variablendeklaration sowie eine Variablenprioritätsvergabe dargestellt sind.

Die zyklischen Programme 3a, 3b umfassen Sprachmittel mit geeigneten Anweisungen und Befehlen, wodurch sequentielle Programme gestartet und Funktionsbausteine parametriert werden. Im einzelnen sind insbesondere folgende Elemente der Sprache innerhalb einer Programmierung des zyklischen Ablaufs verfügbar:

- Operatoren wie beispielsweise Vergleichs- oder binäre Operatoren,
- Standortfunktionen wie z. B. Typwandlungsfunktionen für elementare Datentypen, mathematische Funktionen, binäre Funktionen sowie Funktionen für einen Zugriff auf Systemvariable,
- Standardfunktionsbausteine, z. B. Funktionsbausteine für eine Flankenerkennung, bistabile Funktionsbausteine oder Zähler- und Zeitbausteine, und

- Anweisungselemente in Form von Auswahl-, Wiederhol- und Sprunganweisungen sowie in Form von Steueranweisungen für Funktionen und Funktionsbausteine und Programme.

Die sequentiellen Programme 4a, 4b entsprechen jeweils einer nichtperiodischen Task. Innerhalb der Deklaration wird einem sequentiellen Programm die Priorität der Task zugeordnet. Sequentielle Programme werden von anderen Programmen gestartet und liefern beim Aufruf Rückgabewerte, mit denen sie systemintern verwaltet werden (z. B. Verriegelung gegen mehrfachen Aufruf). Ein Modul kann kein sequentielles Programm, ein sequentielles Programm oder mehrere sequentielle Programme aufweisen. Alle Bewegungsfunktionalitäten sind nur in sequentiellen Programmen verfügbar. Dadurch umfaßt ein sequentielles Programm den Befehlsumfang aller Bewegungsbefehle. Darüber hinaus kann ein sequentielles Programm auch Befehle für eine logische Verarbeitung aufweisen. In den Fig. 5a, 5b, 6, 7a und 7b sind Beispiele von Bewegungsfunktionalitäten gezeigt, wobei in Tabelle 5 allgemeine Bewegungsbefehle, in Tabelle 6 Interpolationsbewegungen und in Tabelle 7 Bewegungsbefehle für einen Master-Slave-Verbund dargestellt sind.

Jedes der zyklischen und sequentiellen Programme 3a, 3b, 4a, 4b umfaßt einen Variablen- und Konstantendeklarationsteil 5, in welchem anwenderspezifische Variablen und Konstanten zu vereinbaren sind. Es werden insbesondere vereinbart:

- Deklaration von lokalen Variablen mit elementaren Datentypen, z. B. ganzzahlige oder reelle Datentypen, Strings,
- Definition von abgeleiteten Datenstrukturen und Bewegungsprofilen,

- Deklaration von Systemvariablen (Achshandle),
- Zuordnung von Variablen zu logischen Geräteadressen,
- Vergabe von Zugriff-rechten für Variable, die für den Datenaustausch bereitgestellt werden,
- Mehrachskonfiguration durch Deklaration unterschiedlicher Achsverbände (Fig. 8),
- Definition von Bewegungsprofilen (Fig. 9).

In den Fig. 8 und 9 sind in Tabellen 8 und 9 Beispiele für eine Deklaration von Achszusammenhängen (Mehrachskonfiguration) und für eine Deklaration von Bewegungsprofilen dargestellt.

Neben der Deklaration von Variablen und Konstanten ist eine Deklaration von Funktionsbausteinen vorgesehen. Bei Anwendung der Funktionsbausteine ist implizit definiert, ob sie beim Aufruf eine schnelle zyklische Task benötigen oder ob sie sich in den Kontext des aufrufenden Programmes einordnen. Funktionsbausteine, die im Kontext des rufenden Programmes laufen, werden innerhalb dieses Programmes instanziert. Schnelle Funktionsbausteine sind innerhalb des Steuerungssystemes hinsichtlich Anzahl und Instanznamen fest vorgegeben. Funktionsbausteine werden periodisch ausgeführt und können mit neuen Parametern versehen werden. Die Ausführung schneller Funktionsbausteine obliegt nicht der Kontrolle der rufenden Task. Somit erfolgt die Ausführung unabhängig von den Regeln der Auswertung des Programmes, in dem der Funktionsbaustein parametriert wurde. Alle anderen Funktionsbausteine laufen im Kontext des rufenden Programmes, d. h., sie ordnen sich in die Reihenfolge der Auswertung der Sprachelemente des Programmes ein.

Zur Verwirklichung von Bewegungsfunktionalitäten sind insbesondere folgende Sprachelemente vorgesehen:

- technologieorientierte Standardfunktionsbausteine (z. B. Nockenschaltwerk),
- Mechanismen für Mehrachskonfigurationen (Konfiguration unterschiedlichster Achsverbände über Achsmodule hinaus zu einem Gesamtsystem),
- bewegungsspezifisch erweiterte (abgeleitete) Datenstrukturen,
- Bewegungsattribute, -funktionen und -befehle.

In den Fig. 10 und 11 sind in Tabellen 10 und 11 Beispiele von wesentlichen Bewegungsattributen und Bewegungsfunktionen dargestellt.

Zur Konfiguration unterschiedlichster Achsverbände über Achsmodule hinaus zu einer Steuerung zum Steuern eines technischen Prozesses und/oder zur Steuerung der Bewegung einer Verarbeitungsmaschine sind Konfigurationselemente vorgebbar. Diese umfassen:

- Ressourcen in Form von Hardwaremitteln,
- Module,

– global Variable,
– Zugriffspfade,

wobei innerhalb einer Konfiguration eine Deklaration von Ressourcen, eine Deklaration von globalen Variablen zur Kopplung von Modulen unterschiedlicher Ressourcen sowie eine Deklarati n von Zugriffspfaden vorgebbar ist. In den Fig. 12 bis 14 sind in Tabellen 12 bis 14 Konfigurationselemente, eine Deklaration von globalen Variablen und eine Deklaration von Zugriffspfaden dargestellt. In einer Ressource selbst werden globale Variable zur Kopplung von Modulen innerhalb dieser Ressource und Module deklariert. Ein Zugriffspfad ist zur Verknüpfung einer Variablen mit einer Eingangs- oder Ausgangsvariablen eines Moduls, zur Verknüpfung einer Variablen mit globalen Variablen einer Ressource oder Konfiguration oder zur Verknüpfung einer Variablen mit einer direkt dargestellten Variablen vorgesehen. Neben einer Deklaration von globalen Variablen für einen Datenaustausch zwischen Modulen und Programmen (einer oder verschiedener Ressourcen) kann ein Datenaustausch über Funktionsbausteine erfolgen. In Fig. 15 sind in Tabelle 15 Beispiele von Kommunikationsfunktionen dargestellt.

Im folgenden wird die Projektierung einer konfigurierbaren Steuerung erläutert. Dazu wird auf Fig. 16 verwiesen, in welcher der Prinzipaufbau einer Rutenwebmaschine dargestellt ist, die zur Fertigung von sogenannten Wilton- und Boucleteppichen geeignet ist. Wesentliche Bestandteile dieser Rutenwebmaschine sind eine Weblade 6, ein Greiferpaar 7 für den Schußfadeneintrag, eine Schaftmaschine, ein Rutenapparat 9, ein Kett- und Polfadenspeicher 10, ein Gewebeabzug 11 und ein Gewebespeicher 12.

Bei der Festsetzung der Eingänge wird grundsätzlich zwischen zeitkritischen und zeitunkritischen Eingängen unterschieden. Zu den zeitkritischen Eingängen werden Wächtersignale (z. B. Schußfadenwächter, Rutenwächter, Stoppsignale etc.) gerechnet, die eine Reaktion der Steuerung in der untersten Zeitebene (IPO-Takt) erfordern. Signale, die die Not-Aus-Funktion der Steuerung auslösen (Not-Aus-Taster, Antriebsüberwachung), werden gesondert verarbeitet. Die übrigen Eingangssignale wie z. B. Bedienhandlungen, zeitunkritische Wächter (Gewebeabzug, Gewebespeicher etc.) werden im Hauptzyklus der entsprechenden Module verarbeitet.

Bei der Festsetzung von Zuständen wird grundsätzlich zwischen folgenden Betriebsbedingungen der Maschine unterschieden:

1) JOG – freies Fahren der Achsen/Antriebe nach Bedienerauswahl,
2) JOG-Referenz – Referieren der Achsen nach Bedienerauswahl oder entsprechend Voreinstellung,
3) AUTOMATIC (Programmabarbeitung):
- stationärer Betriebsfall (Weben),
– Routinen zur Behandlung von prozeß- oder maschinenbedingten Ausnahmesituationen.

Für den stationären Betriebsfall ist von einem Anwender ein technologischer Bewegungsablauf vorzugeben, z. B. ein Bewegungsablauf, wie in den Fig. 17a und 17b dargestellt:

1. Webfach 1 öffnen:
a) Webschäfte in die Raststellung für den ersten Schuß und Weblade in die hintere Endlage bewegen;
2. Schußfaden und Rute eintragen:
a) Bewegen der Greiferstangen in das Webfach,
b) Übergabe des mitgeführten Schußfadens von der linken an die rechte Greiferstange,
c) Rückbewegung der Greiferstangen,

d) Rute in den oberen Teil des Webfaches eintragen;
3. Ansteuerung der Schneid-/Klemmeinrichtung:
a) Abschneiden des Schußfadens und Fixierung bis zum nächsten Schußfadeneintrag;
4. Webfach schließen, Schußfaden und Rute anschlagen:
a) Bewegen der Webschäfte in die Mittelstellung,
b) Weblade in die vordere Endlage zum Anschlagen des Schußfadens und der Rute bewegen,
c) Neupositionieren des Ruteneintrags;
5. Webfach 2 öffnen:
a) Bewegung der Webschäfte in die Raststellung für den zweiten Schuß und Weblade in die hintere Endlage bewegen;
6. Schußfaden eintragen;
7. Ansteuerung der Schneid-/Klemmeinrichtung;
8. Webfach schließen, Schußfaden anschlagen;
9. Fortsetzen im Zyklus (1).

Parallel zum Grundzyklus sind weitere Bewegungsvorgänge zu realisieren:

1. Rutenauszug:
a) Entfernen der letzten Rute vor dem Gewebeabzug und Einschieben in ein Rutenmagazin;
2. Rutenquertransport:
a) Quertransport des Rutenmagazins zwischen den Bewegungen vom Ruteneintrag und Rutenauszug (Erhaltung des Rutenumlaufes);
3. Gewebeabzug:
a) kontinuierlich zur Gewebebildung laufende Nadelwalze;
4. Lieferung von Kett- und Polfäden:

a) kontinuierliche Lieferung von zwei Kettfadensystemen und ein m Polfadensystem;
5. Gewebeaufwicklung:
a) Antrieb des Fertiggewebespeichers.

Darüber hinaus werden vom Anwender ebenfalls die Bewegungsfunktionalitäten der einzelnen Achsen/Antriebe, das Verhalten von Ausgangsgrößen und sonstiger physikalischer Größen gegenüber einer sogenannten Hauptwelle vorgegeben. Im vorliegenden Beispiel werden folgende Ausgangs- und Bewegungsfunktionalitäten vorgegeben:

Achse/Antrieb oder Ausgangsgröße	- Beschreibung	- Parameter
Hauptwelle	- kontinuierlich laufende Rundachse - Masterachse des Systems	- Drehzahl Hauptwelle
Weblade	- mechanisch an die Hauptwelle gekoppelt - Bewegungsfunktion wird mechanisch realisiert	- keine
linker Greifer	- Bewegungsfunktion entsprechend VDI-Richtlinie 2143 für Kurvenscheiben - Polynom 9. Grades	- Greiferweg - Nullpunkt - Winkel der Hauptwelle
rechter Greifer	- linker Greifer	- linker Greifer
Schneid-/Klemmeinrichtung	- digitales Ausgangssignal zur Ansteuerung der pneumatischen Schneid-/Klemmeinrichtung - durch Winkelposition der Hauptwelle bestimmt	- Winkel Hauptwelle für H- und L-Signal
Schaft 1, Polfaden	- Bewegungsfunktion entsprechend VDI-Richtlinie 2143 für Kurvenscheiben - Polynom 3. Grades	- Schaftweg - Nullpunkt - Winkel der Hauptwelle
Schaft 2, Füllfaden	- Schaft 1	- Schaft 1
Schaft 3, Bindefaden	- Schaft 1	- Schaft 1
Speicher Polfaden	- kontinuierliches Abwickeln des Fadenspeichers bei Hauptwellenbewegung - Drehzahl wird zwischen Grenzinitiatoren eingependelt	- Fadenspannung (Grenzinitiatoren) - Motordrehzahl

Achse/Antrieb oder Ausgangsgröße	– Beschreibung	– Parameter
Speicher Füllfaden	– bei maximaler Fadenspannung Abwickeln des Speichers, bis minimale Fadenspannung erreicht ist – Antrieb mit fest eingestellter Drehzahl durch Start-/Stopp-Signal gesteuert	– Fadenspannung (Grenzinitiatoren)
Speicher Bindefaden	– Speicher Füllfaden	– Fadenspannung (Grenzinitiatoren)
Nadelwalze	– kontinuierliche Drehbewegung im Verhältnis zur Hauptwelle – Übersetzungsverhältnis wird durch Parameter bestimmt	– Gewebedichte (technologische Vorgabe)
Gewebespeicher	– Drehbewegung von minimaler Gewebespannung, bis maximale Gewebespannung erreicht ist – Antrieb mit fest eingestellter Drehzahl durch Start-/Stopp-Signal gesteuert	– Gewebespannung im Fertigwarenspeicher (Grenzinitiatoren)
Ruteneintrag	– Bewegung entsprechend den vorgegebenen Winkelbereichen der Hauptwelle – Trapezprofil	– keine
Rutenauszug	– Auszugsbewegung mit konstanter Geschwindigkeit entsprechend den vorgegebenen Winkelbereichen der Hauptwelle – Übergangsprofil ruckbegrenzt	– Geschwindigkeit und Beschleunigung (Fadenklammerung)
Rutenquertransport	– Bewegung entsprechend den vorgegebenen Winkelbereichen der Hauptwelle – Trapezprofil	– keine

Entsprechend dem vorgegebenen technologischen Bewegungsablauf, den vorgegebenen Bewegungsfunktionalitäten der Achsen/Antriebe, dem Verhalten von Ausgangsgrößen und sonstiger physikalischer Größen konfiguriert der Programmierer Software-Module des Steuerprogramms, wobei im vorliegenden Beispiel zweckmäßig mehrere CPU-Einheiten zur Abarbeitung der Module während des Steuerbetriebs vorgesehen sind. Im Beispiel werden folgende Module konfiguriert:

1. Mehrachsmodul 0: Hauptwelle und Greifermechanismus

a) Betriebsartenverwaltung

ADJUST – Routinen zur Behandlung von prozeß- oder maschinenbedingten Ausnahmesituationen,

STATIC – stationärer Betriebsfall "Weben",

b) Auswertung und Umsetzung der Bedienanforderungen,

c) logische Verknüpfung der für den Ablauf erforderlichen Ein- und Ausgänge,
d) Programme zur Beschreibung der Bewegungen der angeschlossenen Achsen (Hauptwelle und Greifermechanismus),
e) Aktivierung der erforderlichen Achsverbände bzw. Einzelachsbewegungen anderer Module,
f) Überwachung von Maschinen- und Prozeßzuständen,
g) Fehlerhandling zum System;
2. Mehrachsmodul 1: Schaftmaschine
a) Auswertung und Umsetzung der Befehlsanforderungen des Mehrachsmoduls 0,
b) Programm zur Beschreibung der Bewegungen der angeschlossenen Achsen (Schaftmaschine);
3. Mehrachsmodul 2: Rutenapparat
a) Auswertung und Umsetzung der Befehlsanforderungen des Mehrachsmoduls 0,
b) Programm zur Beschreibung der Bewegungen der angeschlossenen Achsen (Rutenapparat),
c) Überwachung der Prozeßzustände des Subsystems;
4. Einachsmodul 3: Nadelwalze
a) das Modul enthält kein eigenes Programm,
b) befindet sich in der Betriebsart "azyklischer Befehlsbetrieb" und hat damit ein Befehlsinterface zum Mehrachsmodul 0,
c) über dieses Interface erhält das Modul die Befehle für die Antriebsbewegung mit Angabe der Drehzahl und Drehrichtung;
5. Einachsmodul 4: Polfadenspeicher
a) das Modul enthält das Programm zur Ansteuerung des Polfadenspeichers,
b) Auswertung und Umsetzung der Befehlsanforderungen des Mehrachsmoduls 0,
c) logische Verknüpfung der für den Ablauf erforderlichen Ein- und Ausgänge,
d) Überwachung der Prozeßzustände des Subsystems;
6. E/A-Modul 5: Füll- und Bindekettenspeicher
a) das Modul enthält ein eigenes Programm zur Ansteuerung der Füll- und Bindekettenantriebe (Antriebe werden durch Start-/Stopp-Signale gesteuert, die Drehzahl ist in den Antrieben definiert),
b) logische Verknüpfung der für den Ablauf erforderlichen Ein- und Ausgänge,
c) Überwachung der Prozeßzustände des Subsystems.

Im folgenden wird auf **Fig. 18** verwiesen, in welcher eine Steuerungsstruktur zur Abarbeitung der Module dargestellt ist. Im Beispiel umfaßt die Steuerung ST sechs Teilsteuerungen St0 . . . St5, die jeweils mit einer CPU-Einheit versehen sind und die über einen geeigneten Bus Bu miteinander verbunden sind. Die CPU-Einheit der Teilsteuerungen St0 bearbeitet das Mehrachsmodul 0, die CPU-Einheit der Teilsteuerung St1 das Mehrachsmodul 1. Entsprechend bearbeitet die CPU-Einheit der Teilsteuerung St2 das Mehrachsmodul 2, die CPU-Einheit der Teilsteuerung St3 das Einachsmodul 3, die CPU-Einheit der Teilsteuerung St4 das Einachsmodul 4 und die CPU-Einheit der Teilsteuerung St5 das E/A-Modul 5. An die Teilsteuerungen St0 . . . St5 sind über geeignete Ausgabeeinheiten Ae Antriebe mit entsprechenden Antriebsachsen angeschlossen, welche gemäß den Vorgaben des Software-Module umfassenden Steuerprogramms in Wirkverbindung stehen. Eine Bedien- und Beobachtungsstation BB ist zum Bedienen und Beobachten des technischen Prozesses und/oder des Bewegungsablaufs der Rutenwebmaschine vorgesehen.

Patentansprüche

1. Steuerung, welche versehen ist mit Mitteln zum Steuern eines technischen Prozesses und/oder mit Mitteln zur Steuerung der Bewegung einer Verarbeitungsmaschine und welcher ein Steuerprogramm zuführbar ist, das die Steuerung während eines Steuerbetriebs abarbeitet, **dadurch gekennzeichnet**, daß das Steuerprogramm mit Software-Modulen versehen ist, welche mindestens eine CPU-Einheit der Steuerung während des Steuerbetriebs abarbeitet, wobei die Software-Module derart konfiguriert sind, daß diese zur Prozeßsteuerung und/oder zur Bewegungssteuerung vorgesehen sind.
2. Steuerung nach Anspruch 1, dadurch gekennzeichnet,
   - daß nach Maßgabe des technologischen Bewegungsablaufs der Verarbeitungsmaschine die Anzahl der an Ein-/Ausgabeeinheiten der Steuerung anschließbaren Antriebsachsen und das Zusammenwirken dieser Achsen vorgegeben sind und
   - daß gemäß der Vorgabe der Anzahl der Antriebsachsen und der Vorgabe des Zusammenwirkens dieser Achsen zur Bewegungssteuerung Ein- und Mehrachsmodule konfiguriert sind.
3. Steuerung nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die Software-Module mindestens ein zyklisches Programm und mindestens ein durch das zyklische Programm aufrufbares sequentielles Programm aufweisen, wobei
   - im Falle einer Bewegungssteuerung das sequentielle Programm für die Verwirklichung der Bewegungsfunktionen und das zyklische Programm zur Koordination der sequentiellen Programme vorgesehen ist und
   - im Falle einer Prozeßsteuerung das zyklische Programm zur Verwirklichung von Prozeßsteuerungsfunktionalitäten vorgesehen ist.
4. Steuerung nach Anspruch 3, dadurch gekennzeichnet, daß die Module jeweils versehen sind mit einem Deklarationsteil, auf welchen die Programme des jeweiligen Moduls zugreifen und in welchem Variablen und/oder Datenstrukturen und/oder Bewegungsprofile hinterlegt sind.
5. Steuerung nach Anspruch 3 oder 4, dadurch gekennzeichnet,
   - daß ein Programm mindestens mit einem Funkti nsbaustein versehen ist und

- daß von einem Programm Funktionsbaustein aufrufbar sind.

6. Programmiergerät mit Mitteln zum Erstellen eines Steuerprogramms für ein Steuerung, welche Mittel zum Steuern eines technischen Prozesses und/oder Mittel zur Steuerung der Bewegung einer Verarbeitungsmaschine umfaßt, dadurch gekennzeichnet, daß die Mittel das Steuerprogramm mit Software-Modulen versehen, welche eine CPU-Einheit der Steuerung während des Steuerbetriebs abarbeitet, wobei die Software-Module derart konfigurierbar sind, daß diese zur Prozeßsteuerung und/oder zur Bewegungssteuerung vorgesehen sind.

7. Programmiergerät nach Anspruch 6, dadurch gekennzeichnet,

- daß nach Maßgabe des technologischen Bewegungsablaufs der Verarbeitungsmaschine die Anzahl der an Ein-/Ausgabeeinheiten der Steuerung anschließbaren Antriebsachsen und das Zusammenwirken dieser Achsen vorgebbar sind und
- daß gemäß der Vorgabe der Anzahl der Antriebsachsen und der Vorgabe des Zusammenwirkens dieser Achsen zur Bewegungssteuerung Ein- und Mehrachsmodule konfigurierbar sind.

8. Programmiergerät nach Anspruch 6 oder 7, dadurch gekennzeichnet, daß die Mittel mindestens ein Software-Modul mit mindestens einem zyklischen Programm und mit mindestens einem durch das zyklische Programm aufrufbaren sequentiellen Programm versehen, wobei

- im Falle einer Bewegungssteuerung das sequentielle Programm für die Verwirklichung der Bewegungsfunktionen und das zyklische Programm zur Koordination der sequentiellen Programme vorgesehen ist und
- im Falle einer Prozeßsteuerung das zyklische Programm zur Verwirklichung von Prozeßsteuerungsfunktionalitäten vorgesehen ist.

9. Programmiergerät nach Anspruch 8, dadurch gekennzeichnet, daß die Module jeweils versehen sind mit einem Deklarationsteil, auf welchen die Programme des jeweiligen Moduls zugreifen und in welchem Variablen und/oder Datenstrukturen und/oder Bewegungsprofile hinterlegt sind.

10. Programmiergerät nach Anspruch 8 oder 9, dadurch gekennzeichnet,

- daß ein Programm mindestens mit einem Funktionsbaustein versehen ist und
- daß von einem Programm Funktionsbausteine aufrufbar sind.

11. Anordnung mit mindestens einer Steuerung nach einem der Ansprüche 1 bis 5 und mit mindestens einem Programmiergerät nach einem der Ansprüche 6 bis 10, wobei die Steuerung und das Programmiergerät über einen Bus miteinander verbunden sind.

Hierzu 20 Seite(n) Zeichnungen

Deklarations-richtung		Deklaration	Bemerkungen/ Verweise
Modul		MODUL *Name*: *Modul_Bezeichner* ... (* Modulrumpf *) END_MODUL	– das Bestimmungszeichen ON wird zur Festlegung des Modultypes (*Modul_Bezeichner*) auf logischer Ebene verwendet
Variable	lokale Variable	VAR ... END_VAR	– lokale Variable des Moduls sind für alle zugehörigen Programme global
	Eingangs-variable	VAR_INPUT ... END_VAR	
	Ausgangs-variable	VAR_OUTPUT ... END_VAR	
Programm	allgemeine Deklaration	PROGRAM *Name* (TYPE:= *Typ*, PRIORITY:= *Wert*, INTERVAL:=*Zeitdauer*, SYSSTART := *starttyp*) ... (* Programmrumpf *) END_PROGRAM	– TYPE gibt den Typ des Programmes bzw. der zugehörigen Task an: • NORM = periodische (zyklische) Task • FAST = schnelle zyklische Task • SEQ = sequentielle (nicht periodische) Task – PRIORITY legt die Priorität zum bevorrechtigten oder nichtbevorrechtigten Aufruf der Task fest (*Wert* Typ: UINT (0,1,...,5) – Programme werden zur periodischen Ausführung im angegebenen INTERVALL (*Zeitdauer*) aufgerufen (*Zeitdauer* Typ INT entspricht dem vielfachen der Interpolationstask) – die Angabe des Parameters SYSTART ist nur bei zyklischen Programmen zulässig und legt fest, ob Programme durch expliziten Aufruf (SYSTART:=USER) oder mit Initialisierung des Moduls (SYSTART := INIT) gestartet werden (USER ist voreingestellt)

Tabelle 1: Deklaration von Modulen

FIG 2a

Nummer: DE 197 40 550 A1
Int. Cl.: G 05 B 19/04
Offenlegungstag: 16. April 1998

	zyklisches Programm (ohne festes Zeitraster)	**PROGRAM** *Name* (TYPE:= NORM, PRIORITY:= *Wert*, SYSSTART := *starttyp*) ... (* Programmrumpf *) **END_PROGRAM**	– Programm mit der höchsten Priorität und mit SYSTART:=INIT wird Haupteintrittspunkt des Moduls
	schnelles zyklisches Programm	**PROGRAM** *Name* (TYPE:= FAST, INTERVAL:=*Zeitdauer*, SYSSTART := *starttyp*) ... (* Programmrumpf *) **END_PROGRAM**	– In jedem Modul ist maximal ein zyklisches Programm vom Typ FAST programmierbar
	Programm mit sequentieller Abarbeitung	**PROGRAM** *Name* (TYPE:= SEQ, PRIORITY:= *Wert*) ... (* Programmrumpf *) **END_PROGRAM**	– sequentielle Programme werden ausschließlich über eine explizite Anweisung (CREATE...) gestartet

Tabelle 1: Deklaration von Modulen (Fortsetzung)

FIG 2b

Nummer:
Int. Cl.:
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Nummer: DE 197 40 550 A1
Int. Cl.[redacted]: G 05 B 19/04
Offe[redacted]gungstag: 16. April 1998

Deklaration	Schlüsselwort	Anwendungsbereich/ Bemerkungen
lokale Variable	**VAR**	– Gebrauch innerhalb der Programm-organisationseinheit
Eingangsvariablen (schreibgeschützt)	**VAR_INPUT**	– von außen geliefert, kann nicht in der Pro-grammorganisationseinheit geändert werden
Eingangsvariablen	**VAR_IN_OUT**	– Variable kann im Programm geändert werden
Ausgangsvariablen	**VAR_OUTPUT**	– von der Programmorganisationseinheit nach außen gelieferte Variable
Konstante	**CONSTANT**	– Konstante (kann nicht geändert werden) – Deklaration erfordert Wertzuweisung
Speicherortzuweisung	**AT**	– wird dieses Schlüsselwort nicht angegeben erfolgt eine automatische Zuweisung der Variablen zu einem Speicherort
Ende der Variablen-deklaration	**VAR_END**	– jede Variablendeklaration (unabhängig ihrer Eigenschaft) wird mit VAR_END abgeschlossen
gepufferte Variable	**RETAIN**	– bei Warmstart nehmen die Variablen ihre gepufferten Werte an – bei Kaltstart nehmen die Variablen die vorgegebenen bzw. die im System voreingestellten Initialisierungswerte an
globale Variable	**VAR_GLOBAL**	– werden globale Variable innerhalb eines Konfigurationselemente . Deklariert ist der Geltungsbereich der Variable auf das Element begrenzt indem sie definiert wurden.
Zugriffspfad für Variable	**VAR_ACCESS**	– legt Variable fest, auf die durch die Kommunikationsdienste) zugegriffen werden kann

Tabelle 2: Schlüsselwörter für eine Varaiablendeklaration

FIG 3

Nummer: DE 197 40 550 A1
Int. Cl.[6]: G 05 B 19/04
Offenl gungstag: 16. April 1998

Beispiel	Bemerkungen
VAR *Bit*: **ARRAY [0..6] OF BOOL** := 1,1,0,0,0,1,0; **END_VAR**	– teilt 8 Speicherbits die Anfangswerte zu: *Bit*[0] := 1, ..., *Bit*[7] := 0
VAR *Master* : **INT_AXIS** := *log. Achsadresse*; *Slave* : **AXIS** := *log. Achsadresse*; **END_VAR**	– Deklaration eines Achshandle erfordert Zuordnung zur logischen Adresse der Achse
VAR AT %QX5.1 : **BOOL** := 1; **END_VAR**	– boolesche Variable, direkt adressiert und mit Anfangswert = 1 initialisiert
VAR *Zahl, Wert* : **INT**; *mystring* : **STRING**(10); **END_VAR**	– mehrere Variable gleichen Typs mit Komma getrennt – Zeichenkette mit einer Maximallänge von 10
VAR **CONSTANT** *Wert* : **INT**:= 103; **END_VAR**	– Variable mit konstantem Wert – Konstantendeklaration erfordert gleichzeitige Wertzuweisung
VAR RETAIN *Status* : **ARRAY [0..3] OF INT** := 1,5,0,0; **END_VAR**	– Deklariert als gepuffertes Feld mit den Kaltstart-Anfangswerten *Status*[0]:= 1, *Status*[1]:= 5 *Status*[2]:= 0, *Status*[3]:= 1

Tabelle 3a: Beispiele für eine Variablendeklaration

Fig 4a

Bedeutung	Befehl	Beispiel
Kommunikationspriorität bei gleichzeitigen Zugriff (0-5, 0 höchste Priorität, 3 voreingestellt) Priorität nur für Variablen mit Datenaustausch vorgesehen	% *Priorität* (nicht IEC 1131)	**VAR_INPUT** *Stop* : **BOOL** % 0; *Zahl* : **INT** % 5; **END_VAR**

Tabelle 3b: Vergabe von Prioritäten

Fig 4b

Bewegung		Befehl	Bemerkungen
Referieren	Einachssystem	**REF**	– verschiedene Referiermodi sind über Systemvariablen einstellbar
	Mehrachssystem	**REF** *Achsindex*$_1$,..., *Achsindex*$_n$	– gleichzeitiges Referieren aller Achsen
Positionier-bewegung	geschwindig-keitsgeführt	**POS** (*TYP*$_{opt}$ (*Position*), *Geschwindigkeit*$_{opt}$)	– Einachssystem – Geschwindigkeit aus Systemvariable – *TYP*: Positionsattribut
		POS (*Achsindex*$_1$, *TYP*$_{opt}$ (*Position*), *Geschwindigkeit*$_{opt}$, ... *Achsindex*$_n$, *TYP*$_{opt}$ (*Position*), *Geschwindigkeit*$_{opt}$)	– Mehrachssystem – Achsbewegungen, die innerhalb eines Bewegungsbefehls programmiert werden, starten gleichzeitig
		POS (*Verbundname*, *Achsindex*$_1$ (*TYP*$_{opt}$ (*Position*), *Geschwindigkeit*$_{opt}$)	– Mehrachssystem – Fahren eines Verbundes innerhalb des Positionierbereiches der Masterachse

Tabelle 5: Allgemeine Bewegungsbefehle -Einzelachse und Verbund;

FIG 5a

Nummer:
Int. Cl.:
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

	zeitgeführt	**POST** (*TYP*$_{opt}$ (*Position*), *Zeit*)	– Einachssystem – *Zeit* gibt die Dauer der Positionierbewegung an
		POST (*Achsindex*$_1$, *TYP*$_{opt}$ (*Position*), *Zeit*, ... *Achsindex*$_n$, *TYP*$_{opt}$ (*Position*), *Zeit*)	– Mehrachssystem
kontinuierliche Bewegung	Einzelachsbewegung	**MOVE** (*TYP*$_{opt}$ (*Geschwindigkeit*));	– Einachssystem – Typ: Richtungsattribut
		MOVE (*Achsindex*$_1$, *TYP*$_{opt}$ (*Geschwindigkeit*), ... *Achsindex*$_n$, *TYP*$_{opt}$ (*Geschwindigkeit*))	– Mehrachssystem – wenn Bewegung gestartet und *Geschwindigkeit* erreicht, wird mit Programmabarbeitung fortgesetzt.
	Bewegung im Verbund	**MOVE** (*Verbundname*, *Achsindex*, *TYP*$_{opt}$ (*Geschwindigkeit*))	– nur eine Achse programmierbar stellt den Master des Verbundes dar
	Verbund-bewegung nach externer Masterachse	**MOVE** (*Verbundname*)	– *Achsindex* muß eine externe Achse sein – der Verbund wartet auf die Bewegung der externen Achse (*Achsindex*), um ihr unverzüglich zu folgen
Achsstillstand	Einachssystem	**STOP**	
	Mehrachssystem für Einzelachse	**STOP** (*Achsindex*$_1$, ... , *Achsindex*$_n$)	
	Mehrachssystem für Verbund	**STOP** (*Verbundname*)	– stoppt unverzüglich Achsverbund mit *Verbundname*
		STOP (*Verbundname*, *Achsindex*, *Position*);	– stoppt Achsverbund mit *Verbundname* mit dem Erreichen der angegebenen Achsposition

Tabelle 5: Allgemeine Bewegungsbefehle -Einzelachse und Verbund; (Fortsetzung)

FIG 5b

Nummer: DE 197 40 550 A1
Int. Cl.: G 05 B 19/04
Offenlegungstag: 16. April 1998

		Befehl	Bemerkungen
Interpolation	Gerade	**LIPO** (*Achsindex*$_1$, *Achsindex*$_2$, *Achsindex*$_{3opt}$, *TYP*$_{opt}$ (*Endposition*$_1$), *TYP*$_{opt}$ (*Endposition*$_2$), *TYP*$_{opt}$ (*Endposition*$_{3opt}$), *Geschwindigkeit*)	– Linearinterpolation mit max. 3 Achsen – *TYP*: Positionsattribut
	Kreis im Uhrzeigersinn (positiv)	**CIPO** (*Achsindex*$_1$, *Achsindex*$_2$, *TYP*$_{opt}$ (*Endposition*$_1$), *TYP*$_{opt}$ (*Endposition*$_2$), *Radius*, *Geschwindigkeit*)	
	Kreis gegen Uhrzeigersinn (negativ)	**CIPON** (*Achsindex*$_1$, *Achsindex*$_2$, *TYP*$_{opt}$ (*Endposition*$_1$), *TYP*$_{opt}$(*Endposition*$_2$), *Radius*, *Geschwindigkeit*)	

Tabelle 6:Interpolationsbewegungen

FIG 6

Numm r:
Int. Cl.:
Offenlegungstag:

DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Bewegung		Befehl	Bemerkungen
Masterumschaltung		**SETMASTER** (*Verbundname, Achsindex*)	– angegebender *Achsindex* wird Master für *Verbundname* – Umschaltung kann auch während der Bewegung des Verbundes erfolgen
Verbundmanipulation	Auflösen des Verbundes	**DISABLE** (*Verbundname*)	– alle Achsen des Verbundes können separat verfahren werden
	Wiederherstellen des Verbundes	**RESTORE** (*Verbundname*)	– stellt die zuletzt aktive Verbundkonfiguration her
Synchronisations-bewegungen	Auf-synchronisieren	**SYNCON** (*Verbundname, Slaveindex*)	– synchronisiert eine DEFGEAR-Achse auf eine sich bewegende Masterachse mit maximaler Beschleunigung (Systemvariable)
		SYNCONT (*Verbundname, Slaveindex, Zeit*)	– synchronisiert eine DEFGEAR-Achse auf eine sich bewegende Masterachse in einer vorgegebenen Zeit (impliziert Beschleunigung)
		SYNCONP (*Verbundname, Slaveindex, Profilname*)	– synchronisiert eine DEFCAM-Achse mit einem Einfahrprofil in den Verbund
	Ab-synchronisieren	**SYNCOFF** (*Verbundname, Slaveindex*)	– koppelt eine DEFGEAR-Achse mit maximaler Beschleunigung (Systemvariable) aus dem Verbund – ausgekoppelte Achsen sind separat verfahrbar
		SYNCOFFT (*Verbundname, Slaveindex, Zeit*)	– koppelt eine DEFGEAR-Achse in einer vorgegebenen Zeit aus
		SYNCOFFP (*Verbundname, Slaveindex, Profilname*)	– koppelt eine DEFCAM-Achse mit einem Ausfahrprofil aus

Tabelle 7: Bewegungsbefehle für den Master-Slave-Verbund

FIG 7a

802 016/694

Nummer:
Int. Cl.:
Offenlegungstag:

DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Korrektur-bewegungen	einmalige Korrektur-bewegung auf Slaveachse	**SHIFT** (*Achsindex, Position, Übergangsprofil*)	– Beschleunigen oder Verzögern einer Einzelachse oder des Masters eines Achsverbundes, um eine Positionsverschiebung auf kürzestem Weg (RSP) zu realisieren (**Fehler! Verweisquelle konnte nicht gefunden werden.**) – in Verbindung mit Funktion CHECKPOS ist Druckmarkensynchronisation programmierbar
	Korrektur der Masterposition	**REDEF_POS** (*Achsindex; TYP_{opt} (Position)*)	– die aktuelle oder Sollposition einer Achse wird ohne Bewegung auf eine neue absolute Position definiert – Neudefinition auch während der Bewegung – innerhalb Verbundbewegung kann nur Masterposition neudefiniert werden – Technologie: Bandmarkensynchronisation – TYP_{opt} : Soll- oder Istposition
	Zurücksetzen der Korrektur	**DELETE** (*Achsindex, Korrekturtyp*)	– alle Korrekturen der benannten Achse (*Achsindex*) werden zurückgesetzt
Aussetz-Zyklus	Aussetzen mit Zyklusbeginn	**REST** (*Verbundname, Slaveindex, n*)	– Aussetzen der Slaveachse mit Zyklusbeginn für *n* Zyklen – *n* ist vom Typ: INT
	Aussetzen an definierter Masterposition	**REST_ON_POS** (*Verbundname, Slaveindex, n, Position*)	– ohne Angabe der Position wirkt Befehl wie REST
Einsetz-Zyklus	Einsetzen mit Zyklusbeginn	**INSERT** (*Verbundname, Slaveindex, n*)	– Einsetzen der Slaveachse mit Zyklusbeginn für *n* Zyklen
	Einsetzen an definierter Masterposition	**INSERT_ON_POS** (*Verbundname, Slaveindex, n, Position*)	– ohne Angabe der Position wirkt Befehl wie INSERT – bei vorherigen programmierten Aussetzen muß die gleiche Position verwendet werden

Tabelle 7: Bewegungsbefehle für den Master-Slave-Verbund (Fortsetzung)

FIG 7b

802 016/694

Nummer:
Int. Cl.:
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

	Deklaration	Bemerkungen
Master-Slave-Verbund (positionsgeführt)	*Verbundname* : **DEFCAM** :=*Achsindex*$_1$, *Achsindex*$_2$, ... ,*Achsindex*$_n$;	• Masterachse ist die zuerst in der Deklaration angegebene Achse (*Achsindex*$_1$,) • im Verbund alle Profiltypen zugelassen
Master-Slave-Verbund (Getriebeverbund geschwindigkeitsgeführt)	*Verbundname* : **DEFGEAR** :=*Achsindex*$_1$, *Achsindex*$_2$, ... ,*Achsindex*$_n$;	• Verbund mit Drehzahlgleichlauf • im DEFGEAR-Verbund ist nur der Typ GPROFIL zugelassen • elektron. Getriebe auch über DEFCAM-Verbund möglich (positionsgeführt)
Geometrieverbund (Bahnachsen im kartesischen Koordinatensystem)	*Verbundname* : **DEFGEO** := *Achsindex*$_1$, *Achsindex*$_2$, *Achsindex*$_{3opt}$;	• Interpolationsbewegung nur mit den in DEFGEO deklarierten Achsen möglich

Tabelle 8: Definition eines Achszusammenhangs

FIG 8

Nummer:
Int. Cl.:
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Definitionstyp	Profildeklaration	Bemerkungen
tabellarisch	*Profilname* : **TPROFIL** (*Variable*$_1$, *Toleranz*$_{opt}$)	– *Variable* ist ein im Deklarationsteil definiertes Feld – wenn in einem Verbund ein oder mehrere TPROFIL'e verwendet werden ist für den Master ebenfalls ein TPROFIL als Bezug zu definieren (in der Regel Wertefeld mit konstanter Teilung) – TPROFIL-Achsen in einem Verbund müssen gleiche Felddimension besitzen
geschlossen (vollständiger Zyklus)	*Profilname* : **FPROFIL** (*Bewegungsfunktion*$_1$, *Toleranz*$_{1\ opt}$)	– ermöglicht auch Definition eines elektronisches Getriebes (Bewegungsfunktion = P1)
konstantes Verhältnis	*Profilname* : **GPROFIL** (*Mastergeschwindigkeit*$_1$, *TYP*$_{opt}$ (*Slavegeschwindigkeit*)), *Profilname* : **GPROFIL** (*Masterposition*$_1$, *TYP*$_{opt}$ (*Slaveposition*$_1$))	– Programmierung eines gebrochen rationalen Getriebeverhältnisses – der Verbundtyp der Achse bestimmt ob das Bewegungsprofil drehzahl- oder winkelsynchron ausgeführt wird – Typ: Richtungsattribut gibt an in welcher Richtung die Slaveachse der Masterachse folgen soll
stückweise	*Profilname* : **SPROFIL** [0 .. *Anzahl*] := (*Master_Min*$_1$, *Master_Max*$_1$, *Bewegungsfunktion*$_1$, *Toleranz*$_{1\ opt}$), (*Master_Min*$_2$, *Master_Max*$_2$, *Bewegungsfunktion*$_2$, *Toleranz*$_{2\ opt}$), ... (*Master_Min*$_n$, *Master_Max*$_n$, *Bewegungsfunktion*$_n$, *Toleranz*$_{n\ opt}$);	– nicht geschlossenes Masterintervall zulässig – nicht definierte Bereiche werden mit der Bewegungsfunktion **PO** (Stillstand) ersetzt – stückweise Profilverschiebungen sind programmierbar

Tabelle 9: Profildeklaration

FIG 9

802 016/694

Nummer:
Int. Cl.:
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Nummer:
Int. Cl.⁶:
Off nlegungstag:

DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Bewegungsattribute		
Positionsattribute	Absolut (Linear- oder Rundachse)	*Position* A(*Position*)
	Inkremental (Linear- oder Rundachse)	I(*Position*)
	Absolut in negativer Richtung (Rundachse)	RN(*Position*)
	Absolut in positiver Richtung (Rundachse)	RP(*Position*)
	Absolutposition auf direktem Weg anfahren (Rundachse SP-Shortest Path)	RSP(*Position*)
	Sollposition	COM(*Position*)
	Istposition	CUR(*Position*)
Richtungsattribute	Bewegung in positiver Richtung *Geschwindigkeit* ist immer Absolutwert	*Geschwindigkeit* oder P(*Geschwindigkeit*)
	Bewegung in negativer Richtung	N(*Geschwindigkeit*)
	Sollgeschwindigkeit Istgeschwindigkeit	COM(*Geschwindigkeit*) CUR(*Geschwindigkeit*)
Auswahl des Übergangsprofils einer Achse	Trapezprofil (beschleunigungsbegrenzt)	DYNPROF (*Achsindex* 1)
	ruckbegrenzt	DYNPROF (*Achsindex*, 2)
	parabolisch	DYNPROF (*Achsindex* , 3)

Tabelle 10: Bewegungsattribute

FIG 10

Bewegungsfunktionen	s= Slaveposition φ =Masterpos. oder Zeitbasis	*Funktionsattribut(Parameterliste)*
Stillstand	s=Value	**P0**(*Value*)
konstante Übersetzung	$s=Value_2*\varphi+Value_1$	**P1**($Value_2$, $Value_{1opt}$)
Polynom 2. Grades	$s=Value_3*\varphi^2+Value_2*\varphi+Value_1$	**P2**($Value_3$, $Value_{2opt}$, $Value_{1opt}$)
Polynom 3. Grades	$s=Value_4*\varphi^3+Value_3*\varphi^2+Value_2*\varphi+Value_1$	**P3**($Value_4$, $Value_{3opt}$, $Value_{2opt}$, $Value_{1opt}$)
Polynom 4. Grades	$s=Value_5*\varphi^4+Value_4*\varphi^3+Value_3*\varphi^2+Value_2*\varphi+Value_1$	**P4**($Value_5$, $Value_{4opt}$, $Value_{3opt}$, $Value_{2opt}$, $Value_{1opt}$)
Polynom 5. Grades	$s=Value_6*\varphi^5+Value_5*\varphi^4+Value_4*\varphi^3+Value_3*\varphi^2+Value_2*\varphi+Value_1$	**P5**($Value_6$, $Value_{5opt}$, $Value_{4opt}$, $Value_{3opt}$, $Value_{2opt}$, $Value_{1opt}$)
einfache Sinuslinie	$s=\frac{1}{2}\left[1-\cos\left(Value\cdot\varphi\cdot\pi\right)\right]$	**S0**(*Value*)
geneigte Sinuslinie	$s=Value_1\cdot\varphi-\frac{1}{2\pi}\left[1-\sin\left(Value_2\cdot\varphi\cdot 2\pi\right)\right]$	**S1**($Value_2$, $Value_1$)

Tabelle 11: Bewegungsfunktionen

FIG 11

Nummer:
Int. Cl.
Offenlegungstag:
DE 197 40 550 A1
G 05 B 19/04
16. April 1998

Nummer: DE 197 40 550 A1
Int. Cl.[6]: G 05 B 19/04
Offenlegungstag: 16. April 1998

Deklarations-richtung	Deklaration	Bemerkungen/ Verweise
Konfiguration	**CONFIGURATION** *Name*: ... **END_CONFIGURATION**	– entspricht dem Gesamtsystem
globale Variable	**VAR_GLOBAL** ... **END_VAR**	– die Deklaration von globalen Variablen einer Ressource benötigt die Verbindung zu einer Modulvariablen
Ressource	**RESSOURCE** *Name*: **ON** *Hardware_ID* **END_RESSOURCE**	– eine Ressource faßt Softwaremodule zusammen, die unter einer gemeinsamen Hardware laufen
Modul	**DEFMODUL** *Name*: **ON** *Modul_Bezeichner* *modulvar*: *ressourcevar*; *modulvar*: *direkt. Adresse*; ... **END_MODUL**	– das Bestimmungszeichen ON wird zur Festlegung des Modultypes (*Modul_Bezeichner*) auf logischer Ebene verwendet – im Entwicklungssystem ist eine Beschreibungsdatei enthalten, die jedem *Modul_Bezeichner* ein funktional strukturiertes Software-Modul zuordnet – innerhalb des Deklarationsrumpfes von Modulen werden die Modulvariablen mit Betriebsmitteln (direkte Adressierung) und globalen Variablen der Ressource oder Konfiguration verknüpft

Tabelle 12: Konfigurationselemente

FIG 12

Deklaration	Allgemeine Deklaration
globale Variable einer Ressource	**VAR_GLOBAL** *Name*: *Modulname. Variablenname*: *Typ* ; **END_VAR**
globale Variable der Konfiguration	**VAR_GLOBAL** *Name*: *Ressourcename. Modulname. Variablenname*: *Typ* ; **END_VAR**

Tabelle 12: Deklaration von globalen Variablen

FIG 13

Nummer: DE 197 40 550 A1
Int. [redacted] G 05 B 19/04
Offenlegungstag: 16. April 1998

Allgemeine Deklaration	Bemerkungen
VAR_ACCESS *Name*: *Ressourcename.Modulname.* *Variablenname*: *Typ* : *Zugriff*; **END_VAR**	– *Zugriff* auf Ausgangsvariable eines Moduls – *Typ*: elementarer oder abgeleiteter Datentyp. – *Zugriff*: READ_WRITE oder READ_ONLY
VAR_ACCESS *Name*: *Ressourcename. Variablenname*: *Typ* : *Zugriff*; **END_VAR**	– Zugriff auf globale Variable einer Ressource
VAR_ACCESS *Name*: *Ressourcename. Modulname.* *% log. Speicherort*: *Typ* : *Zugriff*; **END_VAR**	– Zugriff auf direkt dargestellte Variable – *log. Speicherort*

Tabelle 14:Deklaration von Zugriffspfaden

FIG 14

Kommunikationsart	Funktionsbaustein-Aufruf	Bemerkungen
Gerätestatus	*status* := **STATUS**(*Gerät*)	– einem Programm wird der Status des benannten Gerätes (*Gerät*) nach Aufforderung zur Verfügung gestellt – Kommunikationspartner wird über *Gerät* angegeben – der *Status* wird als Wert vom Typ: INT zurückgegeben
Daten lesen	*wert*:=**READ** (*Variablenname*, *Gerät*)	– ein Programm fordert Daten ab – der Zugriff kann von dem Modul, von dem die Daten gelesen werden, kontrolliert werden – *wert* ist lokale Variable, die den Inhalt der gelesenen Variablen zugewiesen bekommt, und muß den selben Typ besitzen wie *Variablenbezeichner*
Daten schreiben	**WRITE** (*Variablenname*, *Wert*, *Gerät*)	– von einem Programm werden die Werte in angebene Variable des Gerätes geschrieben – *wert* muß den gleichen Datentyp wie *Variablenname* besitzen
Programmiertes Melden (nicht quittierbar)	**NOTIFY**(*Ereignis*, *Meldung*, *Gerät*)	– bei Eintreten des definierten Ereignisses (*Ereignis*) können Meldungen (*Meldung*) an das angegebene Gerät (*Gerät*) ausgegeben werden
quittierbar	**ALARM**(*Ereignis*, *Meldung*, *Gerät*, *Quittung*)	– ausgegebene Meldung muß quittiert werden (*Quittung*)

Tabelle 15: Kommunikationfunktion

FIG 15

Nummer: DE 197 40 550 A1
Int. Cl.[6]: G 05 B 19/04
Offenlegungstag: 16. April 1998

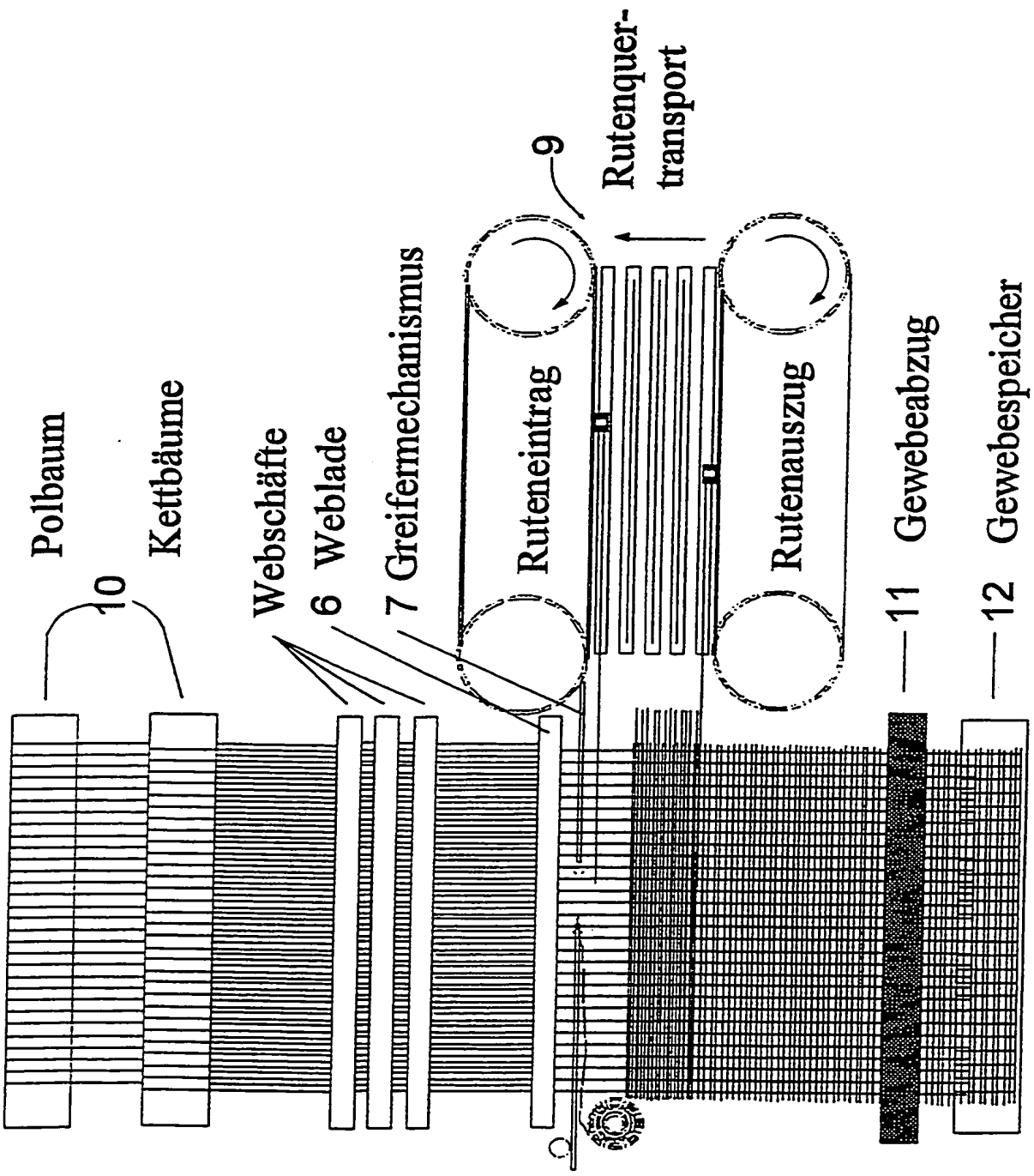
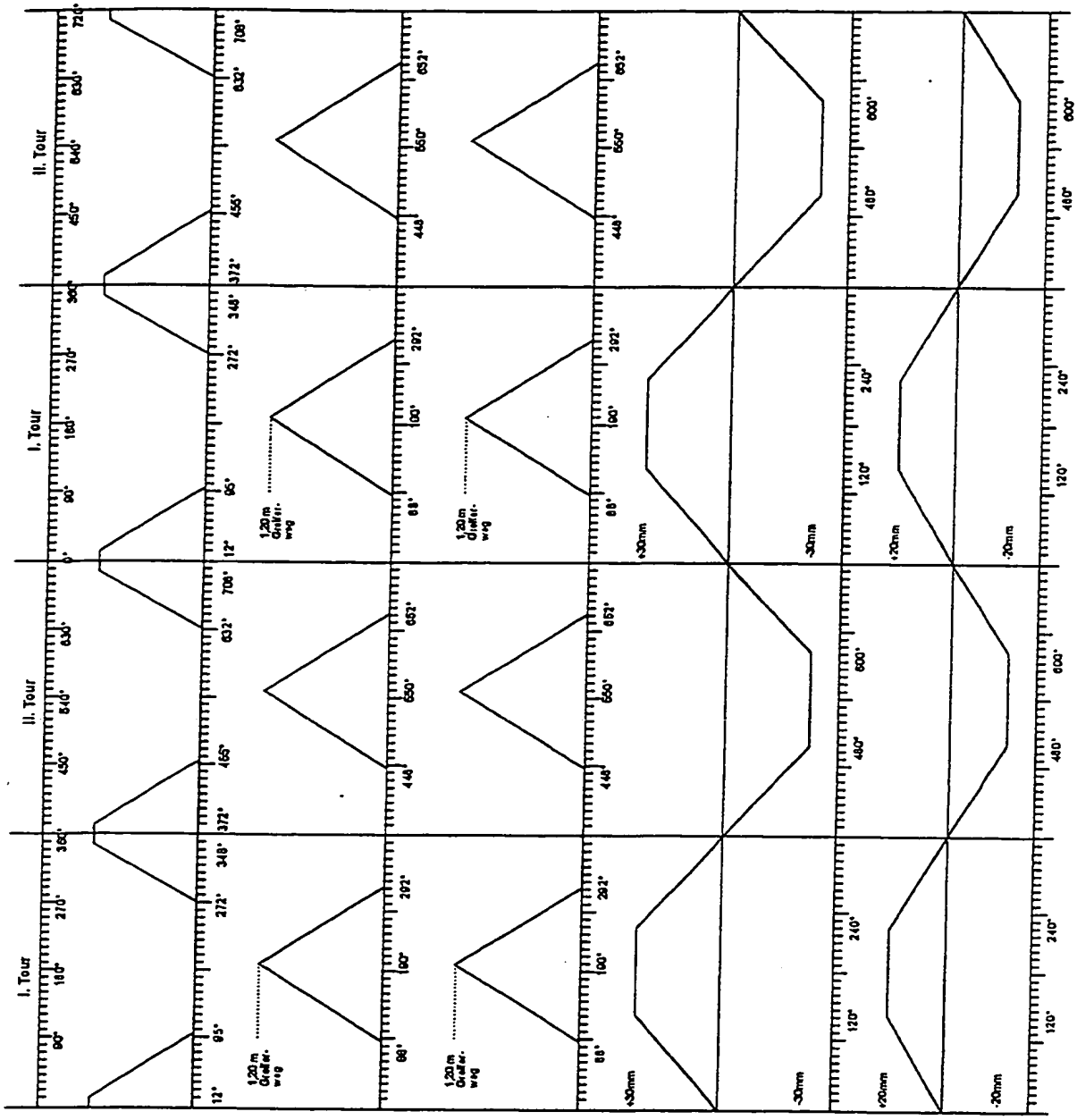


Fig 16

Numm r: DE 197 40 550 A1
Int. Cl[illegible] G 05 B 19/04
Offenl[illegible]ungstag: 16. April 1998



Weblade

linker Greifer

rechter Greifer

Schaft 1 Polbaum

Schaft 2 Füllkette

Fig 17a

Nummer: DE 197 40 550 A1
Int. Cl.[6]: G 05 B 19/04
Offenlegungstag: 16. April 1998

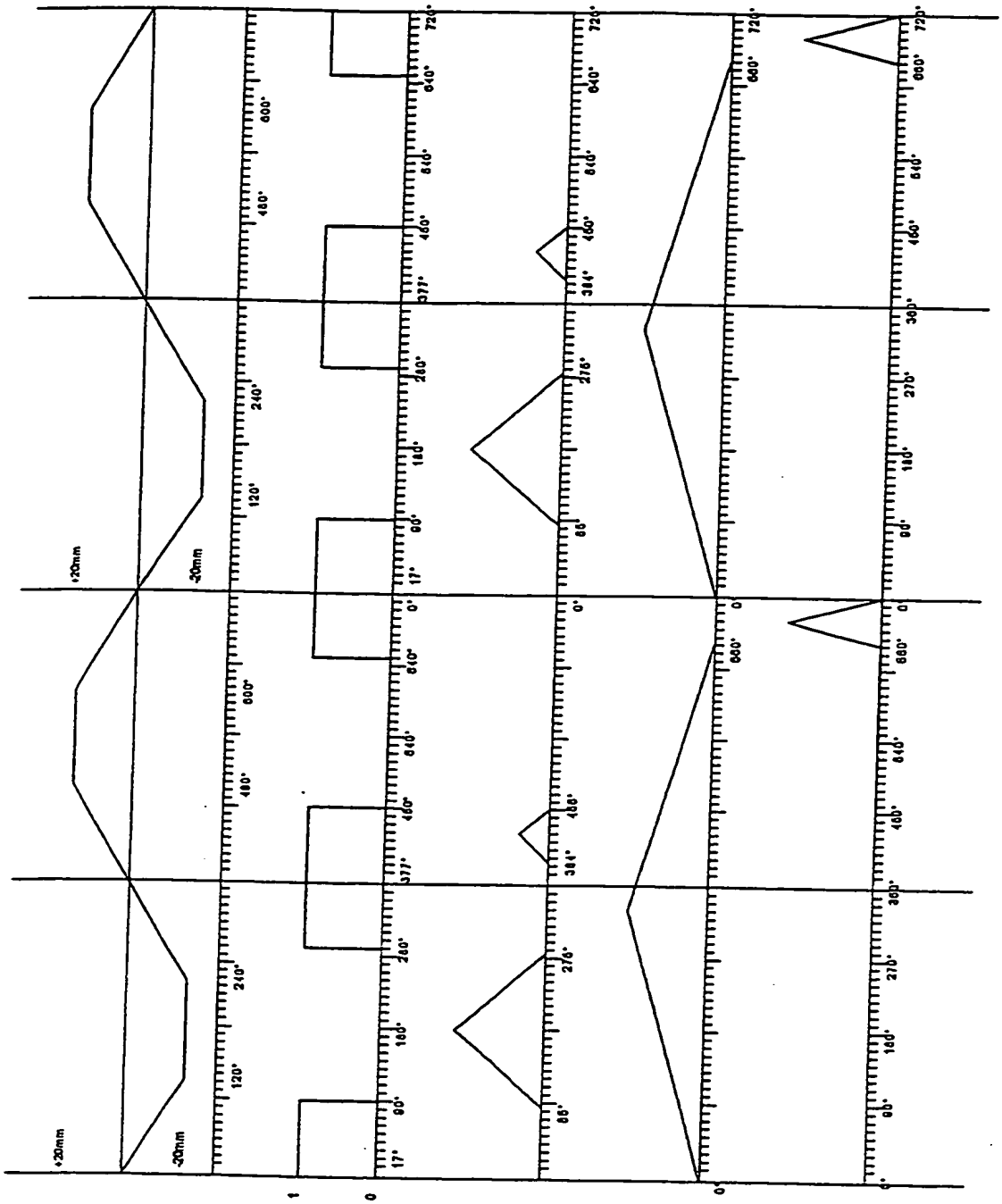


Fig 17b

Nummer: DE 197 40 550 A1
Int. Cl.[illegible] G 05 B 19/04
Offenlegungstag: 16. April 1998

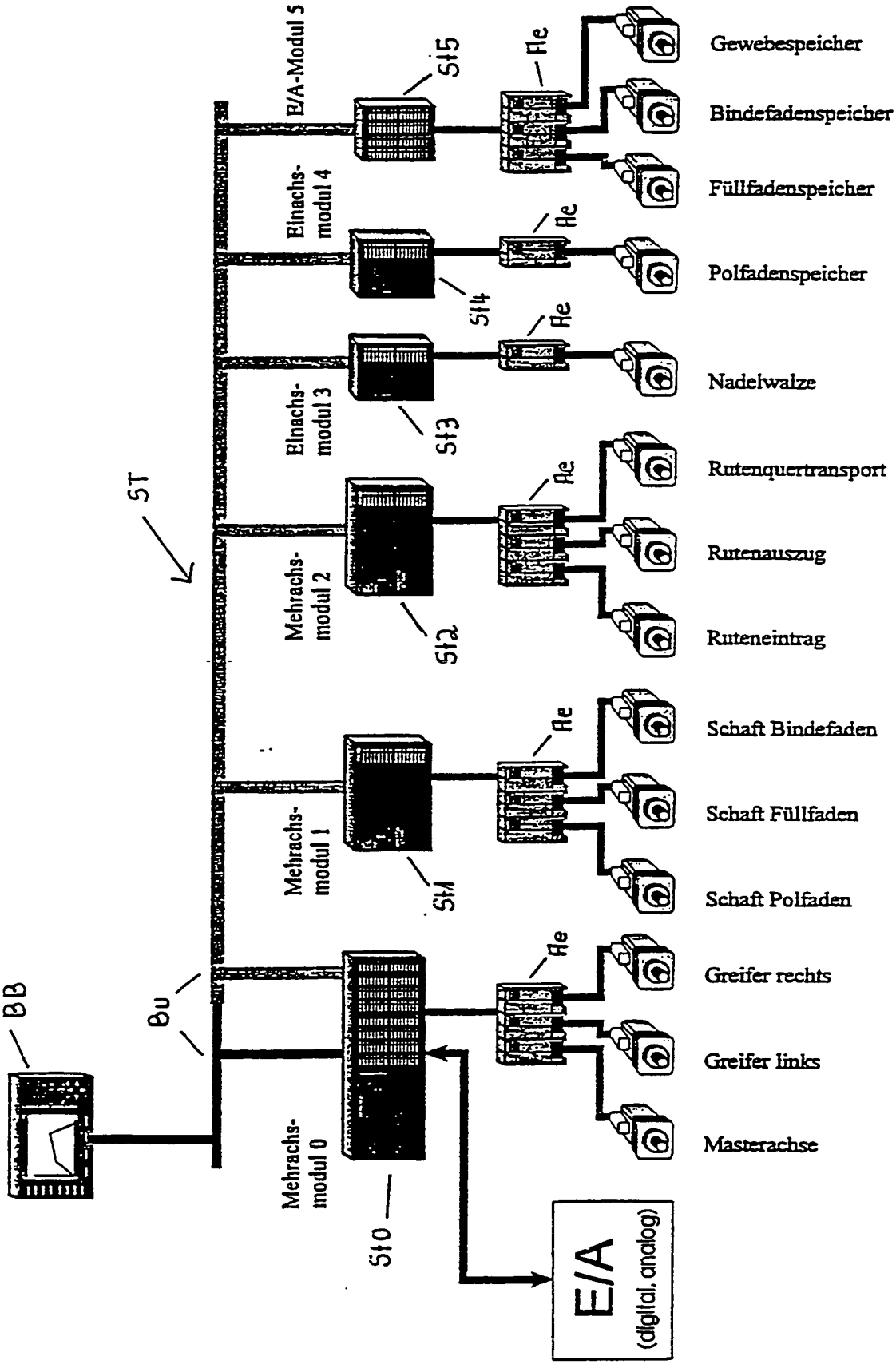


Fig 18

Nummer: DE 197 40 550 A1
Int. Cl.[6]: G 05 B 19/04
Offenlegungstag: 16. April 1998

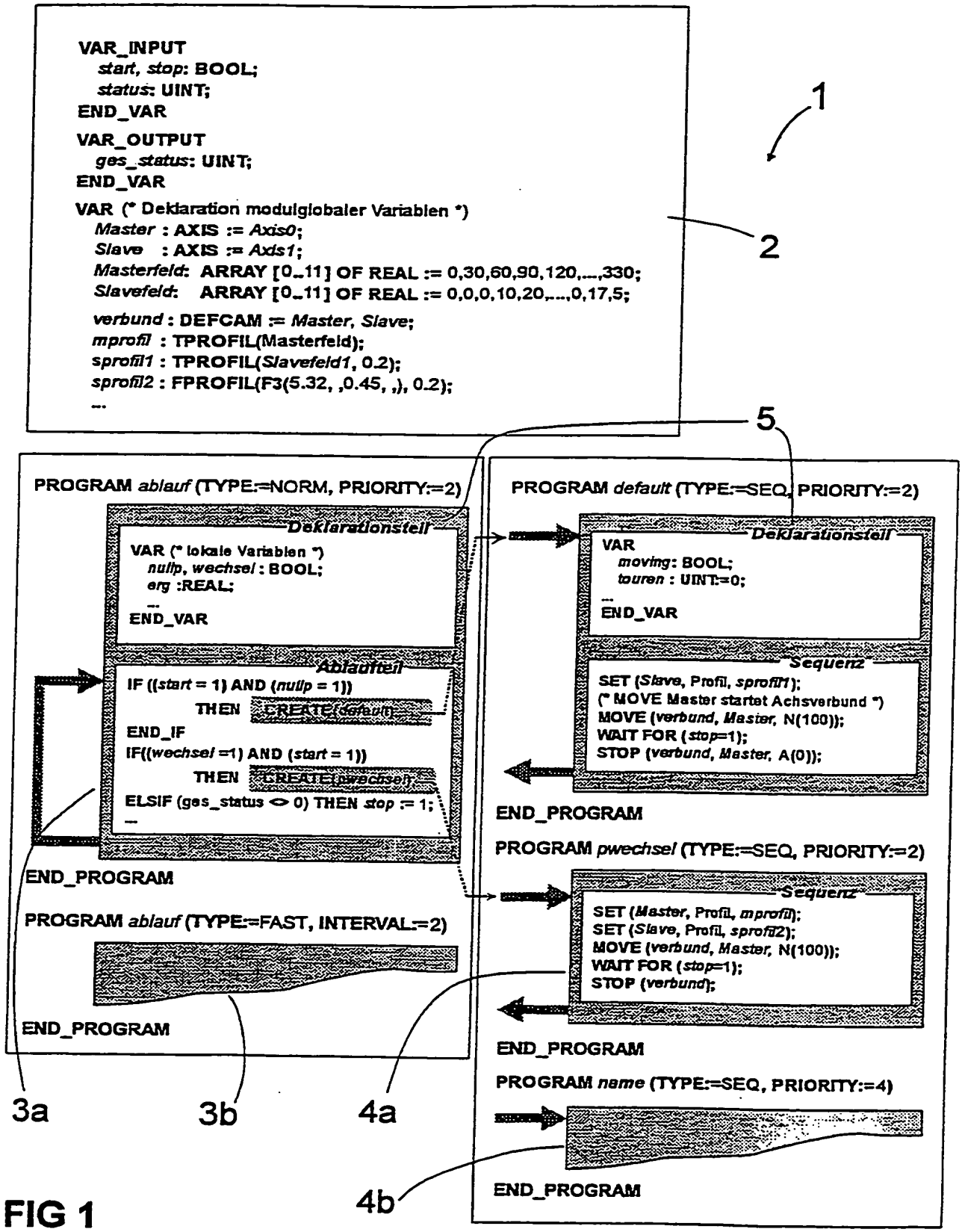


FIG 1